

УДК 621.396.44

ИМИТАЦИОННАЯ МОДЕЛЬ МОДУЛЯ ТАКТИРОВАНИЯ МИКРОКОНТРОЛЛЕРОВ**Усачёв Михаил Валерьевич**

кандидат технических наук, декан факультета информационных технологий Муромского института (филиала) ФГБОУ ВПО "Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых".

E-mail: um.mivlgu@gmail.com.

Адрес: 602264, г. Муром, ул. Орловская, 23.

Аннотация: Рассмотрен подход к моделированию подсистемы тактирования микроконтроллеров с архитектурой ARM Cortex-M0 на примере NXP LPC 1113. Приведено описание подсистемы тактирования, представлена матрица коммутации периферийных модулей и источников тактовых сигналов. Предложен алгоритм оптимизации вызова обработчиков фронтов тактовых сигналов в зависимости от потребностей подключенных периферийных модулей. Для моделирования работы тактовой подсистемы была написана консольная программа на языке C++ для ОС Microsoft Windows XP. Оценка изменения производительности симулятора производилась с помощью измерения машинного времени, затрачиваемого на обработку 1 сек. модельного времени, при неизменной матрице коммутации (за исключением активных фронтов). Результаты моделирования показали возможность увеличения производительности симуляции на 15%.

Ключевые слова: имитационная модель, микроконтроллер, подсистема тактирования.

Введение

Имитационное моделирование микроконтроллеров позволяет оценить работу программного обеспечения в условиях, приближенных к реальным, без операций монтажа и применения дорогостоящего измерительного оборудования. В настоящее время для создания программного обеспечения систем на основе микроконтроллеров используются интегрированные среды разработки. Они различаются по функциональным возможностям и удобству использования. Среди них можно выделить системы, в которых симуляция происходит без подключения внешних радиоэлектронных элементов. Такие системы ориентированы в первую очередь на разработку программного обеспечения и функциональности модели не уделяется достаточного внимания. Поэтому они имеют существенные ограничения и не позволяют симулировать работу многопроцессорных систем. К ним можно отнести IAR Electronic Workbench, Keil uVision, Codevision AVR и др. Среды разработки, ориентированные на моделирование поведения электрической схемы, используют симуляторы на основе Berkeley SPICE 3F5. Они позволяют собрать нужную электрическую схему выбрав требуемые элементы из базы моделей. Здесь можно

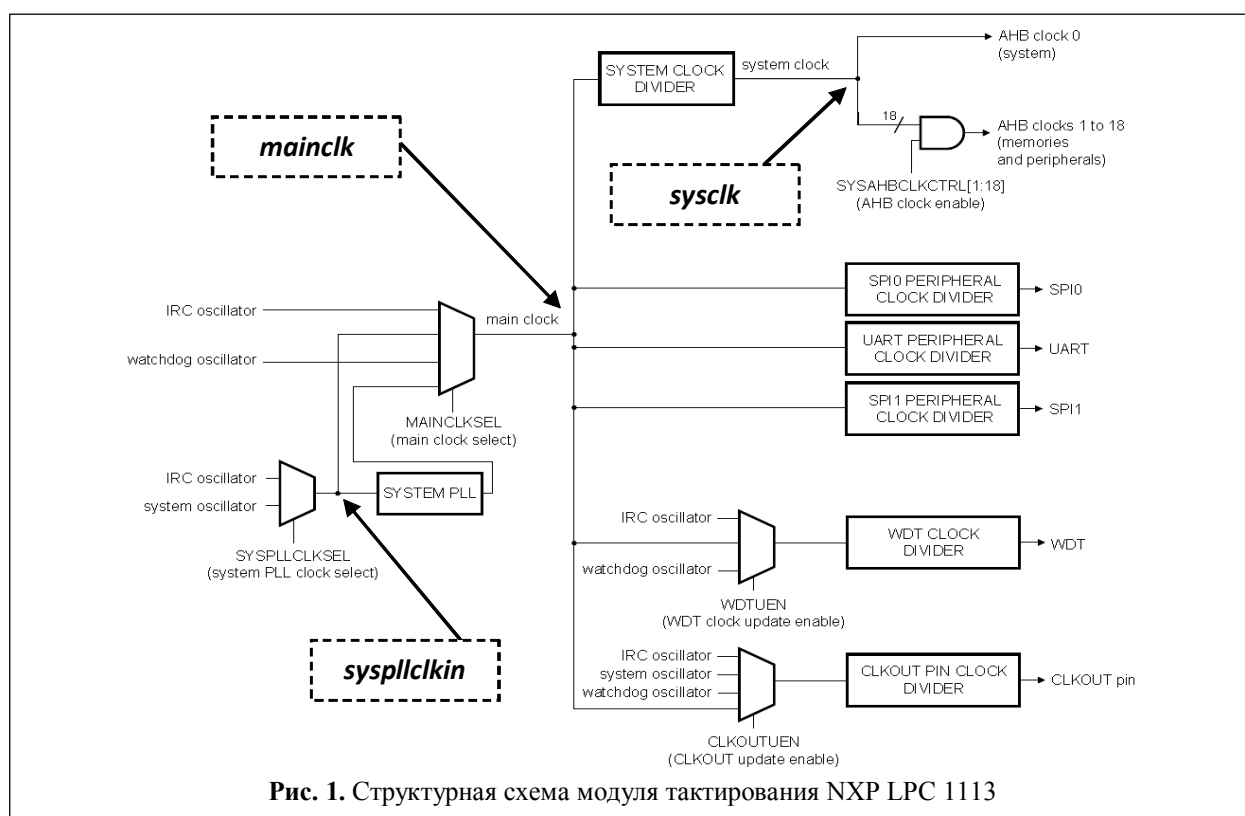
выделить Altium Designer, Labcenter Proteus, NI Multisim и др. [1].

В настоящей статье рассматриваются вопросы разработки имитационной модели подсистемы тактирования микроконтроллеров для интегрированных сред разработки, ориентированных на моделирование.

Современные микроконтроллеры содержат несколько источников тактовых сигналов, причём реконфигурация и подключение к ним периферийных модулей могут осуществляться программно в процессе работы. С увеличением тактовой частоты микроконтроллера растёт число событий, на которые модель должна реагировать. Это приводит к существенному замедлению процесса симуляции.

Целью работы является разработка алгоритма, позволяющего увеличить производительность симулятора за счёт управления источниками тактовых сигналов имитационной модели микроконтроллера, описанной в [1-3].

Рассмотрим структуру модуля генерации тактовой частоты на примере микроконтроллера NXP LPC1113. Он представляет современное семейство микроконтроллеров на архитектуре ARM Cortex-M0. Структурная схема модуля приведена на рис. 1 [4].



Здесь можно выделить четыре аппаратных источника тактовых сигналов, которые могут быть программно настроены на тактирование различных периферийных модулей с помощью записи в конфигурационные регистры необходимых данных. Среди них отметим:

- 1) внутренний RC-осциллятор (IRC oscillator), генерирующий сигнал с частотой 12 МГц;
- 2) осциллятор сторожевого таймера (watchdog oscillator), программно настраиваемый на значение частоты из списка, заданного производителем;
- 3) системный осциллятор (system oscillator), который может быть настроен на использование внешнего тактирования, с помощью входа XTALIN;
- 4) PLL-осциллятор, который также программируется на генерирование сигнала требуемой частоты (при этом он зависит от источника входного тактового сигнала, задаваемого конфигурационным битом SYSPCLKSEL).

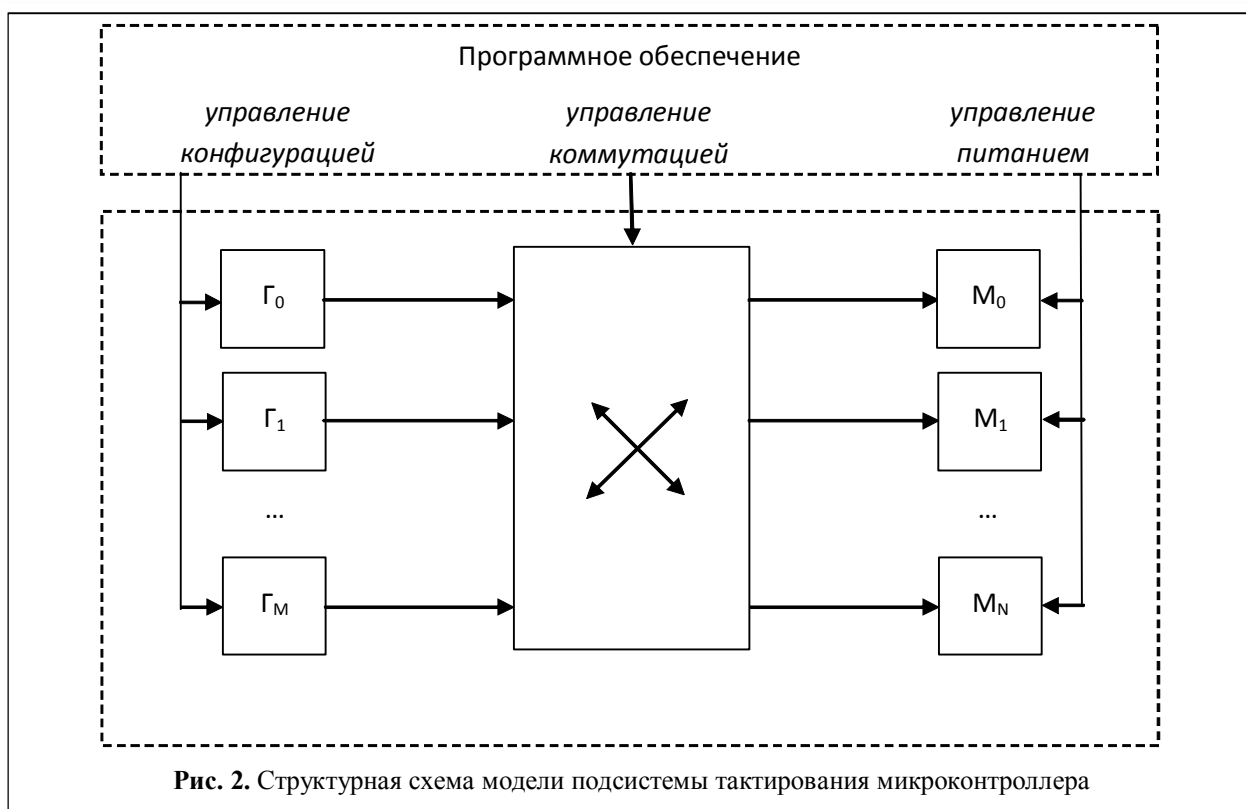
Анализируя функциональные возможности каждого осциллятора, получаем, что модель

абстрактного универсального генератора должна:

- позволять настраивать генерацию сигнала требуемой частоты;
- давать возможность использовать внешние источники тактовых сигналов (т.е. выступать в роли посредника);
- оперативно обновлять свое состояние при изменении списка подключенных модулей.

Структурная схема модели универсальной тактовой подсистемы представлена на рис. 2. Её можно разделить на две части: программную и аппаратную. Настройка параметров, изменение порядка коммутации источников тактовых сигналов, изменение состояния периферийных модулей происходят с помощью программного обеспечения.

В аппаратной части выделим три составляющие: генераторы ($\Gamma_0 \dots \Gamma_M$), системные модули ($M_0 \dots M_N$), а также мультиплексор, отвечающий за коммутацию каждого модуля к выбранному генератору. Системный модуль представляет собой объект, который для корректной работы требует источника тактового сигнала. При этом он должен находиться в



одном из состояний, устанавливаемым программно: включен или выключен. В любой момент времени он должен быть подключен к какому-либо одному генератору. Генератор является объектом, который способен инициировать события в соответствии с частотой, задаваемой в конфигурационных регистрах. Поскольку для тактирования используется цифровой сигнал, то события генерируются для логических состояний «0» и «1». Анализ результатов экспериментов [3] показал, что для некоторых периферийных модулей достаточно осуществлять генерацию события лишь для одного фронта. Это позволит увеличить быстродействие симулятора за счёт избавления от излишних вычислительных операций для неиспользуемого фронта сигнала.

В простом случае мультиплексор представляет собой матрицу коммутации $S[G, M]$, где G – количество генераторов, M – количество модулей. Недостатком такого подхода является необходимость пересчёта всех связей при любом изменении в матрице коммутации.

Введём понятие «ведомого генератора», который обладает функциями приема и передачи сигнала одновременно. Это позволит объеди-

нить генераторы тактовых сигналов в иерархическую структуру и сократить время обработки нового состояния коммутации. На рис. 1 такими генераторами являются: *mainclk*, *sysclk*, *syspllclk*. Поэтому в разрабатываемой модели выбрана матрица $S[K+G, K+M]$, где K – количество генераторов, которые могут быть ведомыми. Генерация событий на самом деле осуществляется только обычными генераторами. Затем события передаются ведомым, которые в свою очередь по цепочке передают их конечным модулям-потребителям.

Может возникнуть ситуация, когда некоторые генераторы будут работать вхолостую, поскольку не имеют потребителей. Она может привести к значительному падению вычислительной эффективности симуляции. Например, если в системе (рис. 1) ведомый генератор *mainclk* будет настроен на источник *system oscillator*, а модули сторожевого таймера и выхода CLKOUT на источник *watchdog oscillator*, то моделирование работы внутреннего RC-осциллятора приведёт к тому, что симулятор создаст и обработает за одну секунду модельного времени 24 млн. событий (суммарно для обоих фронтов), которые по факту являются

бесполезными, поскольку этот генератор не будет иметь подписчиков.

Для предотвращения подобной ситуации в модели используется следующий подход. Каждый абонент (периферийный модуль либо ведомый генератор) содержит описание:

$$D = \{F, S\},$$

где F – перечень фронтов, необходимых для корректной работы абонента; S – текущее состояние абонента, характеризующее способность в данный момент времени обрабатывать сигналы от генератора. При настройке абонентов и переключении их между генераторами производится обновление конфигурационной информации всех заинтересованных вышестоящих генераторов.



Рис. 3. Матрица коммутации

Наиболее удобным является формат матрицы коммутации с учётом состояния абонентов, приведенный на рис. 3. Здесь ненулевые значения в *i*-ой строке и *j*-ом столбце соответствуют подключению генератора G_i к модулю M_j . Интерпретация значений производится исходя из битовой комбинации на основе значений:

- 0x01 – абонент активен;
- 0x02 – абонент ожидает фронт импульса;
- 0x04 – абонент ожидает спад импульса.

Необходимо отметить, что должны выполняться следующие условия:

$$\begin{cases} \sum_{j=1}^{M+K} (C_{ij} \wedge 1) \leq 1, \forall i \in [1..K+G], \\ (C_{ij} \wedge 1) \neq (C_{ji} \wedge 1), \forall i \in [1..K]. \end{cases}$$

Здесь первое ограничение не допускает тактирования модуля от нескольких генераторов одновременно. Второе предотвращает беско-

нечную рекурсию взаимосвязанных ведомых генераторов.

Таким образом, для каждого генератора получаем характеристики с учётом требований абонентов:

- 1) активность *i*-го генератора (A_i)

$$A_i = \left(\bigvee_{j=1}^{M+K} C_{ij} \right) \wedge 1;$$

- 2) необходимость генерации фронта тактового импульса *i*-го генератора (LH_i)

$$LH_i = \left(\bigvee_{j=1}^{M+K} C_{ij} \right) \wedge 2;$$

- 3) необходимость генерации спада тактового импульса *i*-го генератора (HL_i)

$$HL_i = \left(\bigvee_{j=1}^{M+K} C_{ij} \right) \wedge 4;$$

- 4) необходимость генерации сигнала для *j*-го модуля (GE_{ij})

$$GE_{ij} = (C_{ij} \wedge 6) > 0.$$

При изменении коммутации или состояния *j*-го модуля выполняется алгоритм вычисления характеристик вышестоящих генераторов:

- а) при изменении состояния – для всех генераторов при выполнении условия $GE_{ij} = true$;

- б) при переключении на другой источник тактового сигнала – только для текущего и нового генераторов.

Если рассматриваемый генератор является ведомым, т.е. его номер находится в диапазоне [1..K], то для него организуется подобный рекурсивный процесс вычисления характеристик.

Моделирование проводилось на ЭВМ следующей конфигурации: ЦП Pentium Dual-Core E5500 2,8 ГГц; 1,96 ГБ ОЗУ.

Для оценки изменения производительности симуляции используется коэффициент K_{sim} :

$$K_{sim} = \frac{T_{sim}}{T_{mach}},$$

где T_{sim} – модельное время (1 сек); T_{mach} – машинное время, затраченное на симуляцию.

Осуществлялась генерация таблицы коммутации для микроконтроллера NXP LPC1113 в следующем составе:

- 7 генераторов, из них 3 – ведомые;
- 17 периферийных модулей.

Оценивалось время, затраченное только на вызов обработчиков фронтов сигнала (без учёта логики их работы). Был выбран наихудший случай, когда включены все модули.

Результаты, приведённые на рис. 4, показали, что оптимизация вызовов обработчиков тактовых сигналов позволит увеличить производительность симуляции в среднем на 15,2% (без учёта логики работы периферийных модулей).

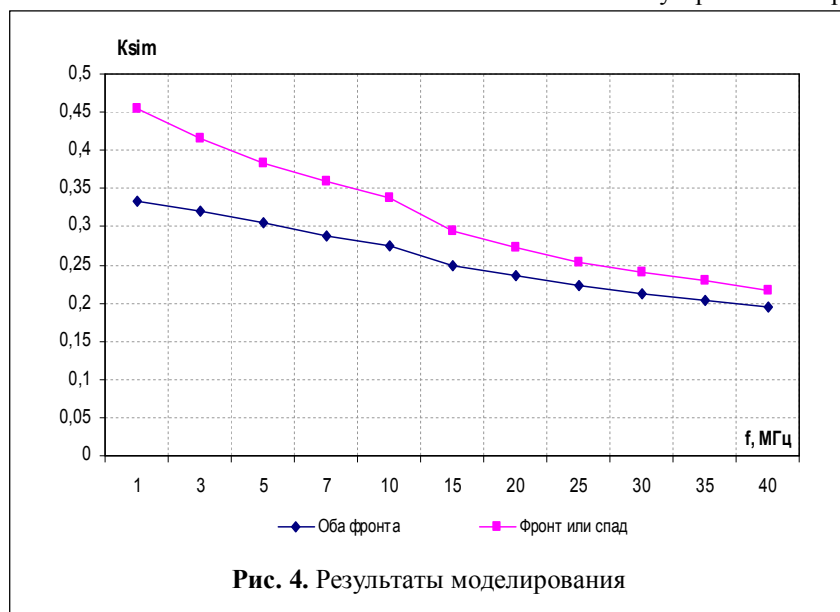


Рис. 4. Результаты моделирования

Таким образом, в работе показан подход к моделированию подсистемы тактирования микроконтроллеров для применения в SPICE-

симуляторах. Он поддерживает множество источников тактовых сигналов и позволяет увеличить производительность симуляции, корректно обрабатывать включение/отключение периферийных модулей и изменение порядка их коммутации.

Литература

1. Усачёв М.В., Мольков Н.П. Обобщённая имитационная модель микроконтроллера смешанных сигналов для SPICE-симуляции // Методы и устройства передачи и обработки информации: межвуз. сб. науч. трудов. – Вып.12 / Под ред. В.В. Ромашова. – М.: «Радиотехника», 2010. – С. 75-78.
2. Усачёв М.В., Мольков Н.П. Разработка имитационной модели ядра ARM Cortex-M3 для интегрированных сред разработки на примере семейства микроконтроллеров NXP LPC 13xx // Радиотехнические и телекоммуникационные системы. – №2, 2013. – С. 69-74.
3. Усачёв М.В., Мольков Н.П. Имитационная модель ядра микроконтроллера с архитектурой ARM Cortex-M3 для SPICE-симуляции // Вестник РГРТУ. – Вып. 44. Рязань: Рязанский государственный радиотехнический университет, 2013. – С.61-66.
4. LPC111x/LPC11Cx User Manual. (дата обращения: 03.05.2014). URL: <http://www.nxp.com>

Поступила 30 мая 2014 г.

English

Simulation model of the microcontrollers clocking unit

Mikhail Valeryevich Usachyov – Candidate of Engineering, Associate Professor, The Dean of the Information Technologies Department Murom Institute (branch) federal state budgetary educational institution of higher professional education “Vladimir State University named after Alexander and Nickolay Stoletov”.

E-mail: um.mivlgu@gmail.com.

Address: 602264, Murom, Orlovskaya st., 23.

Abstract: The article deals with simulation of modern microcontroller clock sources. As a rule, several alternative ways of synchronization signal oscillation and possibility of flexible programmed control in real time are implemented in them. An important feature is that peripheral units can be grouped in a specific way and timed from various sources of synchronization signal. The approach to subsystem simulation of clocking microcontrollers with ARM Cortex-M0 architecture by the example of NXP LPC 1113 is considered here. Peripheral units can be dynamically switched on/off- by the program in the course of operation. Such system behavior does not allow to calculate the moments of switching clock sources and the load of peripheral units on them in advance. It is offered to supervise clock subsystem state adaptively and to eliminate superfluous calculations for syncs

sources reducing thereby computing loading on the simulator kernel. The description of a clocking subsystem on the basis of which the switching matrix of peripheral units and clock signals sources is given. It includes physically present and virtual generators necessary for implementation of a tree-like structure. The algorithm of call optimization of clock signals fronts processors depending on the needs of the connected peripheral units is offered. The console program was written in the C ++ for an operating system of Microsoft Windows XP with the aim of clock subsystem operation simulation. The estimation of a simulator productivity change was made by means of measurement of machine time spent on the processing 1 second of model time, at an invariable matrix of switching (except active fronts). The simulation results showed the possibility of simulation performance by 15 % (excluding the time of synchronization signal front processing by the peripheral units).

Key words: simulation model, a microcontroller subsystem clock.

References

1. *Usachev M.V., Molkov N.P.* Generalized simulation model of a mixed signals microcontroller for SPICE-simulation. Methods and devices of transmission and information processing: Sci. works collection . – Issue 12. Ed. by V.V. Romashov - M.: «Radiotekhnika», 2010. - P. 75-78.
2. *Usachev M.V., Molkov N.P.* Development of a simulation model of ARM Cortex-M3 core for integrated development environments by the example of NXP LPC 13xx microcontrollers. Radiotekhnicheskie i telekommunikacionnye sistemy. №2, 2013, P. 69-74.
3. *Usachev M.V., Molkov N.P.* Simulation model of the microcontroller core with ARM Cortex-M3 architecture for SPICE-simulation. Vestnik RGRU. - Issue 44. Ryazan: Ryazan state radio university, 2013, P. 61-66.
4. LPC111x/LPC11Cxx User Manual. (date of application: 03.05.20142). URL: <http://www.nxp.com>.